# Operation Contracts

## Contract CO1: putHighscore

**Operation:** putHighScore()

**Cross References:** Use Cases: Game is over.

**Preconditions:**  - No possible moves left for the player to make.

- A highcore panel is generated

**Postconditions:**  HighScore is updated in highscore file.

## Contract CO2: saveChanges

**Operation:** saveChanges()

**Cross References:** Use Cases: configureOptions

**Preconditions:**  - Options GUI is successfully loaded.

**Postconditions:** - Changed parameters are successfully loaded into corresponding objects in the environment.

* If user has created a new game piece (lokum) the selected mappings are added to mapping container.
* If user has created a custom game the game engine environment is bind to that instead of our createGame environment.
* A chosen game mode is configured in controller.

## Contract CO3: makeMove

**Operation:** makeMove (pos1, pos2)

**Cross References:** Use Cases: Running game

**Preconditions:**  - Not paused,

-Not game over,

**Postconditions:**  GameEngine gameState is modified.

The new game engine state sent to graphical engine to generate new view.

Graphical Engine created a new view for the user.

## Contract CO4: stopExecution

**Operation:** pauseGame()

**Cross References:** Use Cases: PlayGame

**Preconditions:**  Not already game over.

Not paused.

**Postconditions:**  Game Controller sends pause signals to running objects.

Game Controller gets locked.

## Contract CO5: saveGame

**Operation:** saveGame()

**Cross References:** Use Cases: Play Game

**Preconditions:** a present gameEngine process.

Not game over.

**Postconditions:**  gameEngines current state is saved into an xml file.

None of the objects are changed or manipulated by this method.

## Contract CO6: setGameMode

**Operation:** setGameMode ()

**Cross References:** Use Cases: Configure Options

**Preconditions: -**

**Postconditions:**  A Game Controller is created according to the users preferred game mode.

## Contract CO7: loadGame

**Operation:** loadGame ()

**Cross References:** Use Cases: loadGame

**Preconditions: -**

**Postconditions:**  A saved xml file is read and a previously saved game engine state is created with its according controller.